REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 June 1998	Final 12/18/97 - 6/30/98

4. TITLE AND SUBTITLE
Basic Skills Trainer Software Development

5. FUNDING NUMBERS

6. AUTHOR(S)
Stephen J. Dow

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Alabama in Huntsville
Huntsville, AL 35899

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army Aviation & Missile Command
Redstone Arsenal, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE
Distribution Statement A

13. ABSTRACT *(Maximum 200 words)*

This task continued work performed under previous tasks to enhance software used with training devices to display moving targets overlaid on terrain images. New versions of the software were developed to allow unrestricted terrain geometry, true color images, and 16-bit range data.

19980728 031

14. SUBJECT TERMS
Computer graphics, terrain imagery

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	None

PLEASE CHECK THE APPROPRIATE BLOCK BELOW

DAO# ____________

☐ ____________ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☑ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY; (indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTS (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT F:
FUTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25. WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on ____________ (date) and the AD number is ______________.

☐ In accordance with provisions of DoD instructions. The document requested is not supplied because:

☐ It will be published at a later date. (Enter approximate date, if known).

☐ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

Stephen J. Dow
Print or Type Name

Stephen J Dow
Authorized Signature/Date

256-890-6406
Telephone Number

Basic Skills Trainer Software Development

Principal Investigator and Author: Stephen J. Dow,
Department of Mathematical Sciences
The University of Alabama in Huntsville

Date: June 30, 1998

Contract Name: F/DOD/ARMY/AMCOM/Basic Skills Trainer Software Development

Contract Number: DAAH01-97-D-R005 D.O. 3 Mod 1

1. Introduction

This task continues work performed under two previous tasks: DAAH01-91-D-R005 D.O. 75, "Trainer Methodology Study" and DAAH01-91-D-R002 D.O. 110, "BST Imagery Generation Task." Under those tasks software programs called PathEdit, Depth Finder, and Depth Editor were developed for use in creating scenarios for a training device called the Basic Skills Trainer, or BST, which displays moving targets overlaid on bitmapped terrain images. The reports for the two previous tasks are referenced as [1], [2].

The current task has focused on development of new versions of the earlier programs, with better realism provided by

- Unrestricted terrain geometry in place of the flat terrain model;
- True color terrain images, target colors independent of a terrain palette;
- 16-bit range data for target scaling and occlusion;
- Unlimited number of target views.

The two new programs are called Range Finder and Enhanced Skills Trainer (EST). The Range Finder program is used to create the ground model and range image for a given terrain scene. The EST program is then used to build and run scenarios with moving targets, using the ground model as the basis for target paths and the range image to determine occlusions.

2. The Range Finder Program

2.1 Overview

Range Finder takes as input terrain images and information about the camera used to photograph the terrain and produces as output information about ground coordinates and per-pixel range data. This functionality is roughly a combination of the functionality in the earlier programs Depth Finder and Depth Editor; however whereas those programs work to varying extents with the 8-bit depth data of the existing BST and a flat terrain model, Range Finder works with 16-bit range data and an unrestricted ground model. The main operational steps within the program are:

1. Model setup. Here the term model refers to a single terrain photo or stereo pair of terrain photos, together with associated ground coordinate and range information describing the scene. Camera information required as input is considered part of this setup step. It consists of the set of seven interior orientation (IO) parameters described in section 3.3 of [1], together with the base length (distance between camera locations) when dealing with a stereo model. The camera calibration step which determines the IO is either performed prior to using Range Finder; or a very simple calibration involving computation of a single scaling parameter is performed within Range Finder.
2. Relative Orientation. When a stereo pair is being used, the next step is measuring features which appear in both photos and computing from those measurements the relative orientation (RO) (see section 3.4 of [1]).
3. Ground Point Determination. The user may designate any subset of the points measured on both photos as ground points; i.e. those points which may be observed to lie on the ground. The user may then add additional points by measuring the point on the left photo only and then specifying a range or by editing a 3D position derived from surrounding ground points. If only a single photo is being used, all points are measured in this way.
4. Initial Range Image from Ground Points. Once the set of ground points has been specified, an initial range image is generated which represents the range to the ground at each pixel.
5. Final Range Image including Occluding Objects. Objects projecting above the ground are then "painted" into the range image using tools similar to those in the Depth Editor program.

2.2 File Structure

The files used by Range Finder are described briefly here and in more detail in the sections below. An initialization file (RNGFIND.INI) is used to hold various program settings such as the position and size of the main window; it resides in the same directory as the program executable (RNGFIND.EXE). Then for each model (single image or stereo pair) there are several model-specific files: the original image file or files (.BMP), the model file (.MOD), the ground file (.GND), and the range file (.RNG). These

files may reside in any directory, but that directory must contain all the files for a given model. Since the image files are the initial input, one may consider their location to determine that for the other model-specific files.

2.2.1 Initialization File

The initialization file for Range Finder is an ASCII file named RNGFIND.INI residing in the same directory as the executable file RNGFIND.EXE. It consists of a series of lines of the form "parameter_name = value" which store user preferences and program settings. Here is an example of the contents of RNGFIND.INI:

```
winrect = 53 13 806 708
recent_file = D:\SED\terrains\univ01.mod
recent_file = D:\SED\terrains\Ruth.mod
recent_file = D:\SED\terrains\Benning.mod
brush_size = 1
eraser_size = 9
max_edge_length = 40
edge_percent = 30
shade_val = 0.50
gpt_incr =  0.5
point_centering = on
resource_dir = d:\sed\common\bitmaps
```

The "`recent_file`" values are the model files (described below) most recently opened; these files appear at the bottom of the File Menu for quick access to recently accessed models. The Range Finder program requires several external bitmap files which are used as the symbols on the toolbar buttons; the location of these bitmap files is specified on the "`resource_dir`" line.

2.2.2 Model File

The model file is an ASCII file which stores the parameters and image measurements for the current terrain model. It contains three sections, namely parameters, left_measurements, and right_measurements. The parameters section is similar in form to the initialization file, but contains model-specific parameters. The left and right measurement sections contain the lists of point measurements for the left and right photos; a point measurement being given as an integer point "id" together with its image coordinates in integer subpixel units, a subpixel being 1/256 of a pixel. The point id ties left and right photo measurements together; i.e. left and right measurements for the same point id are measurements of the same real-world object or feature visible in both photos. The id also ties the point to its coordinates in the ground file if the point is a ground point. Here is an example of the contents of a model file:

```
[parameters]
left_image = univ01.bmp
```

```
right_image = univ02.bmp
base_length = 7.315200
io =  375.500  255.761  0.0004600  0.0004600
lens =    0.00350000  -0.00026000   0.00001600

[measurements_left]
  100 80128 84352
  101 98944 40704
  102 126319 67390
  103 122112 41984
  104 154195 84706
  105 130160 106242

[measurements_right]
  100 73472 85888
  101 104960 42624
  102 130560 69120
  103 128128 43776
  104 147712 85504
```

2.2.3 Ground File

The ground file is a binary file which stores the coordinates of ground points in model coordinates, which are 3D coordinates in meters relative to the left camera, where the x-coordinate increases to the right, the y-coordinate increases downward, and the z-coordinate increases as one moves away from the camera. The file format is very simple: an 8-byte header followed by a sequence of 16-byte ground point records. The header consists of four bytes with value 0 followed by a 4-byte integer giving the count of ground points which follow. Each ground point record consists of four 4-byte integers: the point id followed by x, y, and z-coordinates given in centimeters.

2.2.4 Range File

The range file, also referred to as a range image, contains the 16-bit-per-pixel range information. It has the same resolution as the original terrain image, and contains no header or scanline padding. Thus the size of the file in bytes is simply `2*xres*yres`, where the original terrain image has resolution `xres x yres`. Ranges are stored in row-major order starting from the top of the image; i.e. the range for the top-left pixel is first, followed by the remainder of the top row left-to-right, then the second row from the top, and so on. A range R is stored in the file as a 16-bit unsigned integer I using the following scheme: If R is less than 2000 (meters), then I = 10 * R (i.e. I is the range in tenths of a meter). If R is greater than or equal to 2000 but less than 47535, then I = R + 18000. All ranges greater than 47535 are represented by I = 65535. This scheme allows I to represent large ranges (up to 47535 meters at one-meter resolution) while maintaining good resolution at the smaller ranges (one-tenth meter resolution below 2000 meters).

2.3 Concepts and Math Model

2.3.1 Coordinate Systems

The coordinate systems used in Range Finder are the same as in the earlier program called Depth Finder, so some of the following discussion is a synopsis of material from the earlier report [1]. Coordinates within the original image files are called image or pixel coordinates. The 3D coordinate system oriented about the camera, having the camera center of projection at the origin, x-axis to the right, y-axis down, and positive z-axis forward along the line of sight of the camera, is called the camera coordinate system. Units in this coordinate system are meters. Let (x_c, y_c, z_c) denote the camera coordinates of a physical location in the scene. The line joining this point and the origin passes through the plane $z_c = 1$ at $(x', y', 1)$, where

$$x' = x_c/z_c$$
$$y' = y_c/z_c$$

Coordinates (x', y') are called virtual photo coordinates or simply photo coordinates. We can think of these coordinates as measuring where a ray of light coming from an object into the camera intersects an imaginary glass pane placed one meter in front of (the center of projection of) the camera. Virtual photo coordinates are related to image coordinates (x'', y'') through the interior orientation (IO), which is the result of a camera calibration procedure described in [1], [2]. The interior orientation parameters are scale parameters c_x and c_y, translation parameters x_0'' and y_0'', and lens distortion parameters k_1, k_2, and k_3. The equations relating virtual photo to image coordinates are

$$x' = c_x(x'' - x_0'')(1 + k_1 r + k_2 r^2 + k_3 r^3)$$
$$y' = c_y(y'' - y_0'')(1 + k_1 r + k_2 r^2 + k_3 r^3)$$

$$\text{where } r = (x'' - x_0'')^2 + (y'' - y_0'')^2$$

As discussed briefly in section 2.1 above, the IO parameters are computed outside Range Finder, and are regarded as input parameters. If IO parameters are not available, the user can compute a simplified version of the IO within Range Finder in which the scale factors are equal ($c_x = c_y$), $k_1 = k_2 = k_3 = 0$, and (x_0'', y_0'') is assumed to lie exactly at the image center.

After specifying IO parameters, the next step when working with a stereo model is to measure a number of corresponding points on both photos in order to compute the relative orientation (RO) of the stereo pair. The RO consists of 6 parameters, three translation values and three angles, which specify the relationship between the two camera coordinate systems (i.e. the change in location and orientation of the camera in taking the two photos). The internal details of how the RO is computed are discussed in [1] and are transparent to the user in Range Finder, so those details are omitted here. The translation between camera positions is by design mostly a shift in the x-coordinate; and the two camera positions are referred

to as left and right. Under the conventions used here, the coordinate system used for the two-photo model is that of the left camera, so the term model coordinates refers to left camera coordinates. Given the IO and RO parameters and left and right image coordinates of an object or feature visible on both photos, the model coordinates of that point may be computed by stereo reconstruction. The equations used to do this are given in [1] and not repeated here.

2.3.2 Ground Triangulation

The procedure discussed so far produces model coordinates for points measured on both photos; the model coordinates are real-world 3D coordinates in meters. The user designates certain of these points as being ground points; i.e. points which the user visually determines to be on the ground surface. Range Finder places a symbol on the screen over each measured point, and indicates by the symbol color which points are designated as ground points. The ground points form the basis of a triangulated surface; i.e. a surface approximating the ground as a collection of a triangles using the ground points as vertices. At present there is no option to store ground point coordinates in an external coordinate system separate from the model (left-camera based) coordinate system. The use of an external coordinate system based on ground control points may be added in a future version of the software to accommodate the import of GPS or previously collected digital terrain elevation data.

Recalling the description of the camera coordinate system, and assuming the left photo was taken with the camera oriented approximately horizontally, the x- and z-coordinates of ground points may be regarded as planar (horizontal) coordinates and the y-coordinate as the vertical coordinate (elevation). We make the assumption that no two ground points have the same (x, z) coordinates so that there are no perfectly vertical faces in the triangulation. The "top view" display in the software shows the triangulation as seen looking down the y-axis, so that only the (x, z) coordinates of ground points are apparent, with x increasing to the right on the screen and z increasing toward the top of the screen.

After designating certain of the points measured on both photos as ground points, the user may add additional ground points and edit existing ground point coordinates to further specify the ground surface. The software allows points to be added by measuring on the left photo only and specifying a range either by manual keyin or by specifying that the new point lie on the existing triangle on which it falls. It also allows a new ground point to be added on an existing triangle using the top view window. Adding a ground point on an existing triangle does not add any additional undulations to the ground surface; however once a point is added in this manner, its coordinates may be edited (by keyin or interactive adjustment). This allows the user to add features such as ditches or small hills which were not represented in the first pass, but for which elevations can estimated visually. Also, a ground model may be created from a single terrain image (as opposed to a stereo pair) by placing all the ground points using the options just described.

In the remainder of this subsection, we sketch the method used within the software to compute the triangulation. The reader may omit this material without loss of continuity. The triangulation computed is known as the Delaunay triangulation. Computing the triangulation amounts to constructing the lists of

edges and triangular faces using the ground points as vertices. The construction is done entirely in the (x, z)-plane, ignoring the y-coordinate.

The mathematical definition of the Delaunay triangulation is as follows. Let $p_1, p_2, \ldots, p_n$ be n points in the plane, and assume a nondegeneracy condition that no four of the points lie on a common circle. For each point p_i define its Voronoi region V_i as the set of all points in the plane closer to p_i than to any of the other points p_j. Now define edges between pairs of points by the rule that p_ip_j is an edge if and only if the corresponding regions V_i and V_j intersect. It can be proved that these edges form a triangulation; i.e. that they are nonintersecting except at the vertices p_i and that they partition the convex hull of the n points into triangles. The triangulation determined in this manner is the Delaunay triangulation. It possesses the following additional properties,which are used in the construction:

(i) if p_j is the closest vertex to p_i then p_ip_j is an edge;

(ii) if $p_ip_jp_k$ is a triangle then the interior of their circumcircle contains no other vertex.

For proofs of all of these assertions see [3].

The construction now proceeds as follows. The data structure for the triangulation consists of the vertex list (the input points), and the edge and triangle lists to be constructed. The data structure for an edge contains references to its two vertices and to its left and right faces, each of which is either an entry in the triangle list or a special marker indicating the edge is on the boundary of the convex hull of the vertices, which means there is no triangle to that side of the edge. Starting with the first vertex p_1, its nearest neighbor p_i is found. By property (i) above, p_1p_i is an edge, which becomes the first edge in the list. As each new edge is added to the list, we examine in turn the left and right sides of the edge to fill in the references to its two faces. If there is at least one vertex to a given side (half-plane) of the edge, then the edge has triangular face on that side. The third vertex of the triangle is found using property (ii) above. Once the vertex is found, the triangle is added to the triangle list and becomes one of the face references for the current edge. Each of the two other edges of the triangle is added to the edge list, if not already there, to become the new current edge in a succeeding iteration. The algorithm stops when there are no new edges to be examined.

2.3.3 Constructing the Range Image

Once a set of ground points sufficiently modeling the elevation contours has been determined, it is time to begin constructing the range image. Range Finder treats the operations of measuring points and editing range data as separate modes of the program, and changes the menus and toolbar when switching between the two modes. Details of the operation of each mode are described in the following section, but a conceptual overview is included here. Although it is possible to switch back and forth between the two modes, inconsistencies in the range data can result, and things work smoother when the ground model is completed first, and then the range image constructed without going back and changing the ground model. The range editing mode resembles a "paint" program, in which regions of the image are selected, and then range values are painted into the current selection.

If a range image for the current model has previously been saved, the process starts with that range image; otherwise a range image is initialized with constant value of the maximum representable range, which is 47535 meters. That value may be left in the final results for the portion of the image representing sky or distant hills on the horizon. The first step for the user is to then fill in the remainder of the image with ground ranges. A menu command ("Set Ground Ranges") allows any given selected region to be filled with ranges taken from the ground model. When this command is executed, each triangle whose projection to image coordinates intersects the selected region is scan converted to generate a range for each pixel. The algorithm to do this keeps track of those pixels which have already been assigned a range in this manner, so that if the pixel is revisited in scan-converting another triangle the pixel is assigned the new range only if it is smaller than the previously assigned range. In this way each pixel is assigned the range of the closest triangle overlapping that pixel.

After the ground ranges have been set, the user paints in ranges for objects above the ground, using tools discussed in more detail in following section.

2.4 Program Operation

2.4.1 Creating or Selecting a Model

On startup Range Finder reads its initialization file RNGFIND.INI, which contains the names of the most recently opened model (.MOD) files. These files are listed at the bottom of the File menu, and the most recent one, if one exists, is opened. If this is the not the model the user wants, then the user needs to select another model file from the list or select one using the Open command on the File menu or create a new model using the New Model command on the File menu. The New Model dialog box is shown in Figure 1. Because the image file or files must exist prior to setting up the model depending on it, its location will determine the directory for the model. Thus the easiest way to fill in the New Model dialog box is to start by hitting the Browse button beside the Left Image field and traverse to the image file for the left photo. Selecting that image file will fill in the top three fields in a manner similar to that shown in Figure 1. The model name defaults to the base name of the image file but may be changed if desired. For a stereo model, the right image and base length (distance between camera positions) in meters must be entered. The left and right image files must reside in the same directory. Hit the OK button to proceed to the IO initialization which uses the dialog box shown in Figure 2. When first

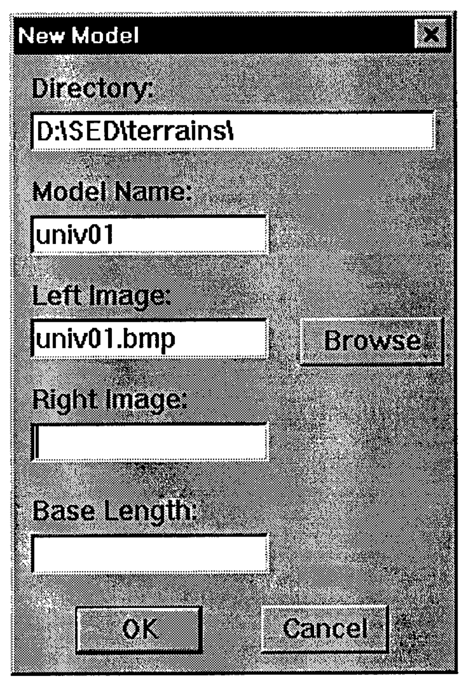

Figure 1: New Model Dialog Box

displayed the IO parameter fields are fillled with their most recent previous values, and the “Use Given IO” option is set. The user may edit the values and press okay to accept the displayed values, or select the “Compute Photo Scale” option. The latter option fills the IO parameter fields with values for a simple scaling transformation for which the translation parameters (x_0'', y_0'') are half the image size, the x and y scale factors are equal ($c_x = c_y$) and set to a default value, and the lens distortion parameters k_1, k_2, and k_3 are zero. After the OK button is hit, a second dialog box comes up (See Figure 3), which is used to correct the common scale factor. The user is directed to measure a distance on the (left) image using the mouse. This means to place the mouse cursor over one end of an object whose size is known or can be estimated, press and hold the left mouse button down to drag a line to the other end of the object, and release the mouse button. The software then fills a field showing the object size in pixels. The user must fill in two other fields to tell the software the size of the object in meters and its range in meters. Hitting OK after this sequence will change the common scale factor (c_x and c_y) to a value computed from these inputs. There is also an “Edit IO” command under the Dialogs menu which allows the user to change the IO parameters after having created the model. The Edit IO dialog box also has a button labeled “Compute Photo Scale” which performs the same function as selecting that option on the New Model IO dialog box.

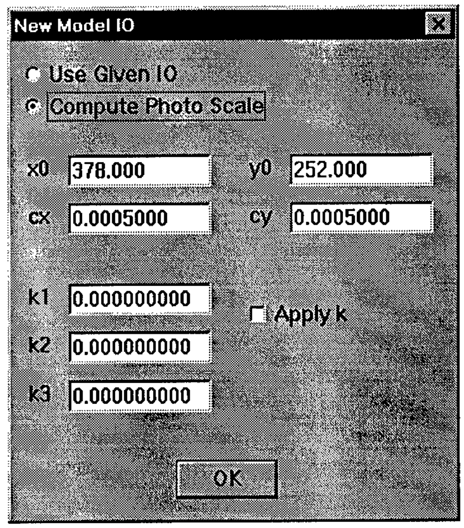

Figure 2: New Model IO Dialog Box

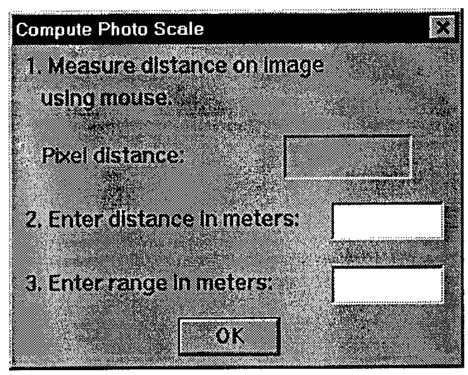

Figure 3: Compute Photo Scale Dialog

2.4.2 Measuring Points

Once a model has been created and the IO parameters entered, the next step is to measure points for RO, in the case of a stereo model, or to head straight into measuring ground points in the case of the single-photo model. By measuring a point we mean specifying pixel coordinates for a point by clicking the mouse with the cursor over the desired location in the image. Figure 4 shows the Range Finder main window as it appears in measurement mode with both left and right photos displayed along with the Point List dialog box. There is a menu item under Dialogs to control display of the point list; it is also displayed when the user selects the Measure Point command on the menu or toolbar. The point list dialog lists all points which have been measured on one or both photos, as well as the active point, which may not yet have been measured. The active point is the highlighted point in the list, whose id is shown in the field

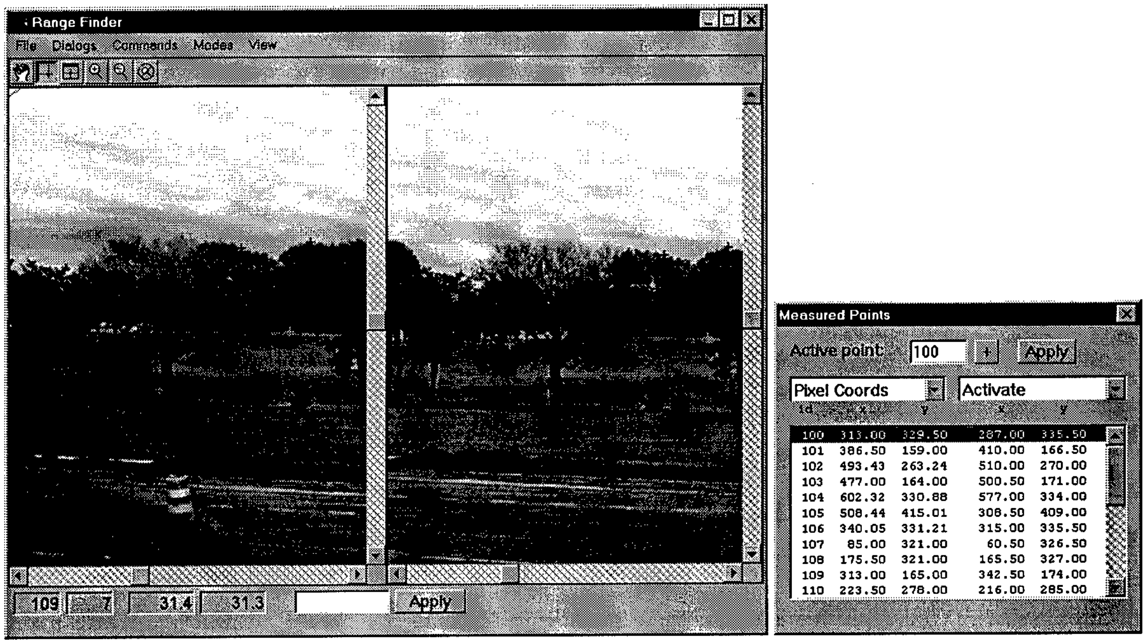

Figure 4: Range Finder Main Window in Measurement Mode

near the top of the dialog, and is the point currently being measured. Measured points are indicated by a cross symbol in the image windows, with the active point being indicated by a diagonal rather than vertical cross. The active point may be changed in several ways: (1) selecting a row in the point list or control point list, (2) keying in a new point id and hitting the apply button, (3) hitting the plus button, or (4) using the grab point command. The plus button causes the point id of the active point to increment to the next larger integer; it is convenient for measuring several points with sequential point ids. The grab point command is activated under the Commands menu or on the toolbar; once activated if the left mouse button is hit with the cursor near a measured point, the closest such point becomes active. If Active Point Centering (under the Options menu) is turned on, then each time a previously measured point becomes active the image or images on which it is measured scroll so as to center the measured point in the display window. Once a point is active and the Measure Point command is active, hitting the left mouse button with the mouse cursor in the image display window causes the current mouse cursor position to be interpreted as the point's measurement. If it was previously measured, that measurement is updated. The coordinates are displayed in the point list. Typically in measuring a point the user will want to scroll the image or images to the area of interest, then use the Zoom In command to allow precise positioning of the mouse cursor.

Once at least 6 points have been measured on both photos, the software will attempt to compute relative orientation. The underlying mathematics of this operation are given in [1], and those details not needed to use the software are omitted here. The results of the relative orientation computation are a set of 6 parameters consisting of 3 rotation angles (about each axis) and 3 translation components (along each axis), which represent the position and orientation of one camera relative to the other. With these

parameters and a measurement on each photo of a common physical point, the software can compute the model coordinates of that point. A pulldown field on the point list dialog allows the user to switch from pixel coordinates to model coordinates, and the software automatically attempts to compute RO when the user takes this action. There is also a menu item to compute RO. The RO computation also takes place each time the user "withholds" or "reinstates" a point from the list of points used in the computation, or deletes a point. Points currently withheld are indicated in the point list by a "W" to the right of the point's model coordinates. The pulldown field at the upper right of the point list dialog indicates the action taken when a given point is selected by clicking on it. The default action is "Activate," meaning make the point the active point. The other two options are Withhold/Reinstate, meaning toggle the withhold state of the point, and Delete, meaning delete the point's measurement(s) on whichever photos it has been measured on.

The user should examine the z component of the model coordinates, which represents range, on the point list to determine if the RO computation is resulting in reasonable range values. Suspect points may be withheld if the computation fails. After good RO results are obtained, additional points may be measured on both photos to serve as ground points. If these adversely affect the RO results, the points may be withheld (but may still be designated as ground points as described in the next section).

2.4.3 Specifying the Ground Model

Once the RO is satisfactory and enough points have been measured on both photos, it is time to specify the ground points which are to be triangulated into a ground model. At this point the user does not really need to display both photos and may wish to use the Left Only option under the view menu so that only the left photo is displayed. The actions for specifying the ground model are (i) designating which of the points having model coordinates are to be considered ground points, (ii) placing additional ground points, and (iii) editing ground coordinates.

The first step is to designate which of the points previously measured on both photos represent features deemed to lie on the ground, using the Designate Ground Points command. When this command is in effect (indicated by a check beside the menu item), the user clicks the mouse near a point symbol to toggle its ground status. Initially all the measured points are indicated with red symbols meaning they are not ground points; ground points are indicated by green symbols. Also ground points are indicated by a "G" to the right of their coordinates on the point list.

Once some ground points exist, the user may display the ground triangulation over the image display using the Show Triangulation item under the Modes menu. Also a separate window showing the triangulation looking down the y-axis may be displayed using the Show Top View item under the View menu. The triangulation is automatically updated internally as ground points are added or changed, but at present the displays do not always update to show the new triangulation, so the user may need to use the Repaint menu item or toolbar button.

There are several ways to place additional ground points. First, a point measured on the left photo only may be converted to a ground point by specifying its range. This is accomplished by making it the

active point, typing a range (in meters) into the keyin field at the bottom of the main program window, and hitting the Apply button beside that field. This automatically converts it to a ground point. Second, a new point may be simultaneously measured on the left photo and its range specified using the Place Ground Point command. When this command is in effect and the mouse is clicked over the left photo within the area which has been triangulated, a point measurement is taken and a range determined by placing the point on the triangular face containing it. The third method of added a ground point is to click the mouse within a triangle in the top view window, with the Measure Point command active. The new point's x and z coordinates are determined by the cursor position, and its y coordinate is determined by the triangle in which it falls. In all of these commands it is important to make sure the correct point id is active (as indicated in the field at the top of the Point List dialog and on the Edit Ground Point dialog).

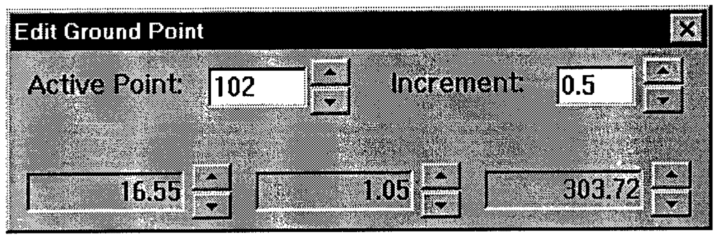

Figure 5: Edit Ground Point dialog box

Any of the ground points, those with model coordinates from RO as well as the others, can have their ground coordinates edited using the Edit Ground Point dialog shown in Figure 5. The full list of ground points/coordinates can be displayed on the point list dialog by setting the pulldown field for Ground Coordinates. When the field is set for Model Coordinates the list shows only coordinates generated from the RO parameters. It is possible for a point to have different coordinates under those two listings. The coordinates listed under Ground Coordinates are the ones saved in the GND file.

2.4.4 Creating the Range Image

An overview of the process of creating the range image is given in section 2.3.3 above. As discussed there, this part of the software operates as a separate mode, with its own menus and toolbar buttons. To enter this mode, select Edit Range Data under the Modes menu. To return to the mode where points are measured, select Measure Points under the Modes menu.

When editing the range image, only the left photo is shown. The view menu has options to display the image only, the range data only, or to overlay the range data over the image data. The general procedure for most operations is to "select" a region of the image and then apply the operation (usually assigning a range value) to this selection. The currently selected region is shown using a special color called the selection color, which the user can change (Set Selection Color under the Dialogs menu). The selection color is stored in the Range Finder initialization file for use in future sessions.

The toolbar as it appears in range edit mode is shown in Figure 6. The leftmost six buttons on the toolbar correspond to the first six items under the Select menu: Mark Pixels, Erase, Mark Line, Mark Rectangle, Fill Region, and Mark Edge. These are used in the manner of painting tools to draw pixels into, or erase them from, the current selection. The next four buttons are used to control the thickness of lines

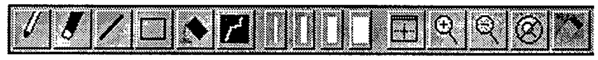

Figure 6: Toolbar for Range Editing

drawn or erased (Mark Pixels, Mark Line and Erase commands). The next four buttons are used to control the portion of the image visible in the window. The leftmost of these is used to center a given point in the window; when this command is active and the mouse is clicked within the image area, that location is shifted to the window center. The next two buttons are used to zoom in or out, maintaining the current location at the center of the window. The next button is used to return to the default view position and one-to-one zoom. The rightmost button on the toolbar is used to clear the current selection; i.e. return to the state where no pixels are selected.

The first step in creating the range image is to select the whole area below the horizon (perhaps using the rectangle tool) and then apply the Set Ground Ranges command (under the Process menu). As discussed in section 2.3.3 this scan converts the ground triangles into pixel-by-pixel range values. After applying this command, the user must assign ranges only to occluding objects above the ground. The object is selected, either by direct painting or using one of the additional tools described below. A range is then entered into the keyin field at the bottom of the window and assigned to the selected pixels by hitting the Apply button. The information in the other readouts at the bottom of the window may be useful in determining ranges. The first two readouts show the x and y pixel coordinates of the current mouse position, the next readout shows the range to the ground at the current mouse position, and the fourth readout shows the current range for that pixel within the range image. If the object currently being assigned a range extends to the ground, the range to apply can be obtained using the Get Range From Selection item under the Select menu. That command finds the range currently assigned to the bottommost pixel of the current selection, and fills that range into the keyin field. If that appears to be the correct range to apply, the user can hit the Apply button. Note that hitting the Apply button both applies the range to the selected pixels and clears the selection.

To avoid the need to reselect complex objects, a system of storing and retrieving selections is available under the File menu. The menu items Read Selection File and Write Selection File are fairly self-explanatory; a compact file format using run-length encoding is used to store the selected region. The user is allowed to store the file in any convenient directory. When retrieving a selection with the read command, the selection from the file is merged with any currently selected pixels. Thus if a large object has been stored in several parts, the parts can be retrieved one by one to form a selection of the whole object.

The first three items under the Process menu (Define ROI, Collect Sample Statistics, and Select by RGB) are all essentially part of a single command used to select objects which can be isolated from their surroundings by color values. Here are the steps to follow:

1. Select some representative pixels, using whatever paint tools are appropriate.
2. Choose the Collect Sample Statistics menu item. This command examines color values for the currently selected pixels and saves the information internally.

3. Select a larger, containing region and then choose the Select ROI menu item. The ROI (region of interest) is the set of pixels which will be examined in the final step to determine which pixels belong to the object. Thus it should include all pixels of the object, but should be fairly closely cropped around the object to prevent unwanted pixels from being included.
4. Choose the Select by RGB menu item. This will bring up the dialog box shown in Figure 7. The ranges for red, green, and blue are those which were collected in step 2. Hitting the Apply button causes pixels in the ROI with color values within the specified ranges to be selected. The ranges can be manually adjusted if this is helpful in getting the proper pixels selected.

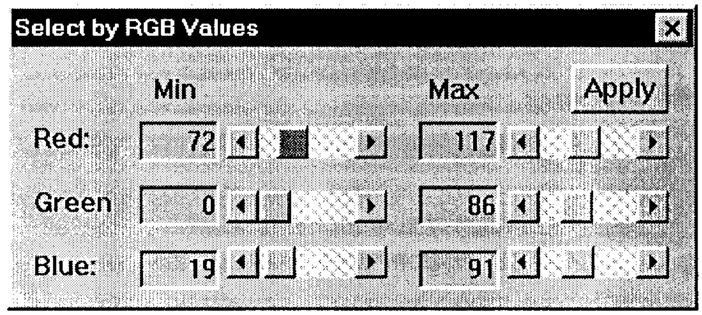

Figure 7: Select by RGB dialog box

Another tool which can be used to aid in selecting complex shapes is the Mark Edge command. When this command is active, and the user clicks the mouse in the image window, the program determines if the cursor location is on an edge (where color values change rapidly) and if so attempts to follow the edge for a given number of pixels. If an edge is found the pixels along the edge are selected. Two parameters can be adjusted on the Edge Options dialog box shown in Figure 8. The top scrollbar controls the maximum number of pixels along the edge to follow and select in one activation of the command (i.e. after each mouse click). The bottom scrollbar controls a threshold parameter which the user can adjust within the range 1 – 99; this parameter is a percentage of the image pixels to be regarded as edge pixels which can be followed. Thus smaller values restrict the edge follower to very well-defined edges and larger values allow it to pick up more poorly-defined edges (and in the process possibly pick up false edges). The two settings for the Edge Follower are stored in the Range Finder initialization file.

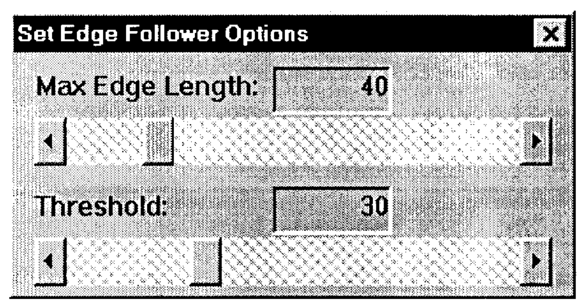

Figure 8: Edge Options dialog box

3. The Enhanced Skills Trainer Program

3.1 Overview and File Structure

Once a terrain model has been set up using Range Finder, scenarios are built and run using the Enhanced Skills Trainer (EST) program. The term scenario refers to the combination of a terrain model and a set of targets following prescribed paths over the terrain. EST contains two major submodules, a path editing module used to build scenarios and a runtime module used to run them in full screen mode as needed for display during training. The latter module has not yet been integrated to communicate with a hardware training device, but some work in that direction has been done. The intention is to expand the full screen portion of the software to incorporate communication with the training device, display of additional status icons, scoring, and other user interaction needed to make the software operational as a trainer. Exiting full screen (or training) mode would take the user back to a windowed environment to review results, change scenarios, or, as in the current software, to edit paths and create new scenarios.

3.1.1 Initialization File

As with Range Finder, whose initialization file RNGFIND.INI was described earlier, the EST program reads an initialization file named EST.INI at startup. Here is an example of the contents of that file:

```
winrect = 10 10 873 688
full_screen = 640 480 16 100 60 440 360
image_dir = \sed\terrains
target_dir = \bst\targets
path_dir = .
resource_dir = d:\sed\common\bitmaps
```

As with Range Finder, "`resource_dir`" specifies the location of bitmap files used as toolbar symbols. Some of the bitmap files are used by both Range Finder and EST.

3.1.2 Data File

The EST data file is an ASCII text file named EST.DAT which resides in the same directory as the the EST executable file EST.EXE. It lists the available terrains, targets, and scenarios and lists the paths making up each scenario. Here is an example of the contents of EST.DAT:

```
[TERRAINS]
UNIV01   u01_   375.500   255.761 0.0004600 0.0004600 0.0035000 -0.0002600 0.0000160
RUTH     rut_  2048.000   256.000 0.0003000 0.0003000 0.0000000 0.0000000 0.0000000
BENNING  ben_  2048.000   256.000 0.0005435 0.0005435 0.0000000 0.0000000 0.0000000

[TARGETS]
BMP2       0.05000
BTR60      0.04500
T72        0.04800
T80        0.05200
```

```
[SCENARIOS]
>FIELD                 UNIV01     4000
   0200 T80
   0201 T72
   0202 T80
>RANGE1                RUTH       24000
   0101 BMP2
   0102 T72
>benn1                 BENNING    24000
   0100 BMP2
```

Interpretation of the file contents is given in the following sections.

3.1.3 Terrain, Range, and Ground Files

Each line in the terrains section of the EST data file corresponds to an available terrain for which there should be a terrain image file (.BMP), a range image file (.RNG), and a ground file (.GND). All three of these files have the same base name, which is given in the first field of the line in the EST data file. The current software requires this base name be no longer than eight characters. The three files must reside together in the directory specified in EST.INI as the value of `image_dir`, which defaults to ..\terrains. The second field in a terrain line of EST.DAT is the terrain id, which is a four character abbreviation of the terrain name used in constructing path file names. The remaining seven fields of each terrain line contain the seven IO parameters described in section 2.3.1 in the following order: c_x, c_y, x_0'', y_0'', k_1, k_2, and k_3.

3.1.4 Target Files

At present the same format is used for the target files as used in the BST Path Editor program. A target file consists of a concatenation of bitmap files representing the various angular views of the target. The name of this concatenated file is the name of the target type followed by the extension IND. For example the name of the file for the T72 tank is T72.IND. The available target files are listed in the reside in the directory specified in EST.INI as the value of `target_dir`, which defaults to ..\targets. The first field in each line in the targets section of EST.DAT is the target name; the second field is a scale factor specifying the number of meters representing by each pixel in the target bitmap files.

3.1.5 Scenarios and Path Files

The scenarios section of EST.DAT lists the available scenarios, using the "greater than" symbol (>) to indicate the start of each scenario. That symbol must be the first non-whitespace symbol on a line, and is followed on the same line by the scenario name, the terrain name, and the number of frames (sixtieths of a second). Following that line are separate lines for each target appearing in the scenario; each of those lines contains a path id and a target type. The path id is a four character identifier used as part of the path filename, specifically the path file name is the 8.3 character sequence consisting of the terrain id followed by the path id followed by the extension .PTH. In the example given, the first path file of the first scenario listed would be named U01_0200.PTH. The path files reside in the directory specified in EST.INI as the value of `path_dir`, if one is specified, and otherwise in the same directory as the image

files. A path file is an ASCII file specifying a series of path vertices, one to a line. The format of each line of the path file is

```
x y z view_num frame_ct
```

where the first three fields are decimal values of the ground coordinates of the vertex (in meters) and the last two fields are integer values for the view number of the target to be used between this vertex and the next and the number of frames (sixtieths of a second) alloted to the segment between this vertex and the next. The fields must be separated by at least one space character, but are otherwise unrestricted in width and spacing.

3.1.6 Target Path Concepts

At present, target paths in EST always lie on the ground; the basic structure of the program allows for flying targets, but there is no convenient way yet to construct paths for flying targets or to handle continuously changing target views to simulate helicopter rotors as in the existing BST. Target paths are based on the ground model constructed for the terrain in Range Finder. As indicated by the format of path files described in the previous section, a path is determined by a sequence of vertices given in ground coordinates. Within EST source code, these vertices are called primary vertices and the portion of the path joining consecutive primary vertices is called a path segment. In EST a path segment is not necessarily a straight line as it is in the BST Path Editor, because the two vertices may lie on different faces of the ground triangulation. This is handled in EST by constructing a straight line segment in the xz plane joining the (x, z)-coordinates of the two primary vertices, finding all intersections of this line segment with edges of the triangulation (when viewed as triangles in the xz plane), and placing intermediate vertices at these intersections. The path segment then consists of the line segments along the actual elevated triangles joining these intermediate vertices. When editing paths in EST the user moves the locations of the primary vertices only; the locations of the intermediate vertices are recomputed each time this happens.

A primary vertex is classified as a static vertex if it the following primary vertex is at the same location. Static vertices are used to cause a target to sit at a given location and possibly change views while there. A scenario has a total frame count set by the user which determines how long the scenario runs. The next to last vertex in each path is always a static vertex used to consume the remaining frames of the total frame count not used up by the earlier path segments.

The existing BST system associates with each path a sequence of events, which record changes to the status of a target as it moves along the path. The functionality for dealing with events has been copied from the BST Path Editor into the EST program without change, using the same file format and user interface. Because that portion of the program has already been documented in the earlier report and also because this functionality is likely to undergo significant change in future software, no further discussion of events is included here.

3.2 Program Operation

3.2.1 Initial Windows

As mentioned earlier, EST operates in two modes, path edit mode and full screen mode. The program always starts in path edit mode, and an item on the "Run" menu allows the switch to full screen mode. In full screen mode there are no window borders, menus, etc present; a message on the screen tells the user to hit the escape key to exit full screen mode. Figure 9 shows the initial program windows

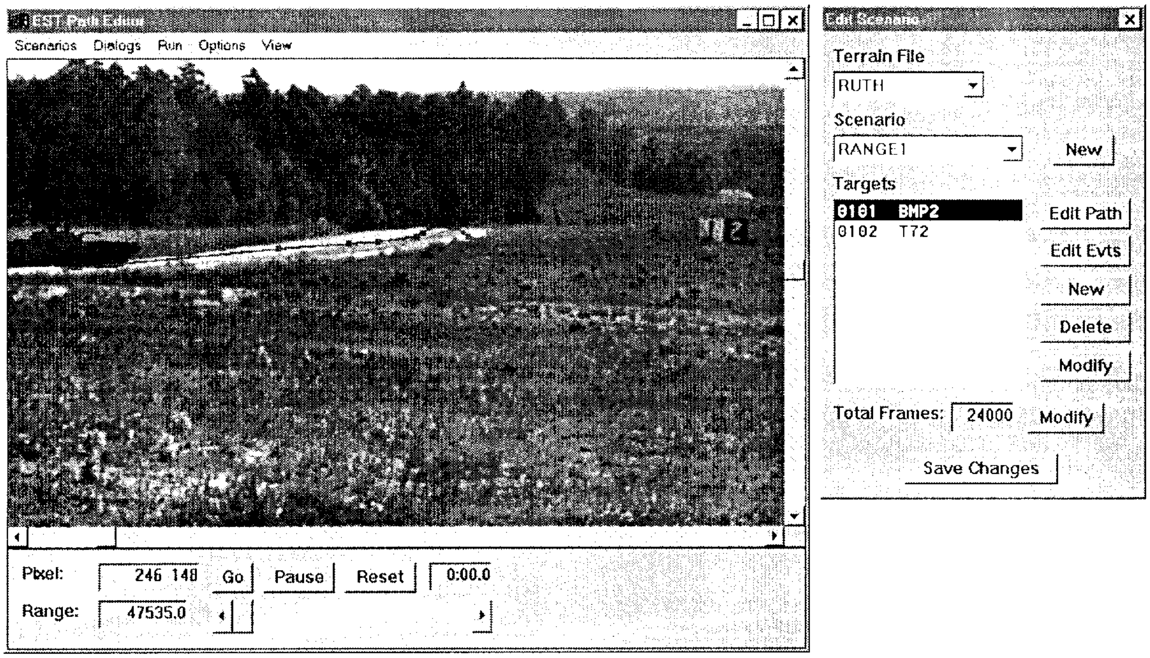

Figure 9: Initial EST Windows

displayed by EST at startup. On the left is the main program window, which serves to display the terrain image and overlaid targets and active target path. It also has the main program menu at the top and some readouts and controls at the bottom. Closing this window exits the program. On the right is the "Edit Scenario" dialog box. On startup the main window will be blank, indicating no terrain file or scenario has been loaded; this will also be reflected by the top two fields of the Edit Scenario dialog box being blank. A terrain may be loaded by selecting from the drop-down terrain list, or a scenario loaded by selecting from the drop-down scenario list. Loading an existing scenario automatically loads its associated terrain. To create a new scenario, first select a terrain, then hit the button labeled "new" next to the scenario field.

Further discussion of the edit scenario dialog box is given in the next section. We now briefly describe the controls at the bottom of the main program window. At the left are readout fields for pixel coordinates and range; these readouts update as the mouse cursor moves over the terrain image displayed above. The range displayed is the value stored in the range image at the indicated pixel. The controls to the right of the readouts are the current frame scrollbar and associated go, pause, and reset buttons, and

timer readout. These control and display information about the scenario timeline. Whenever a scenario is loaded within EST, the user can run or pause the scenario, using the go and pause buttons or corresponding options on the run menu. The reset button returns the scenario to the initial frame. The scrollbar extents correspond to the length of the scenario, so that dragging the scrollbar button to the left or right end of the scrollbar takes the user to the initial or final frame of the scenario. The timer readout shows the current position along the timeline in tenths of a second.

3.2.2 Editing a Scenario

Once a scenario has been loaded, the targets for that scenario are listed on the Edit Scenario dialog box (see Figure 9). Each line in the target list shows a path id and a target name. The target highlighted in the target list is called the active target. If the "Draw Active Path" option is turned on (Options menu), then the path of the active target is drawn in red in the main display window.

The buttons to the right of the target list on the Edit Scenario dialog are:

(1) Edit Path	This button closes the edit scenario dialog box and brings up a dialog box for editing the path of the active target.
(2) Edit Evts	This button closes the edit scenario dialog box and brings up a dialog box for editing the events along the path of the active target.
(3) New	This button adds a new target to the scenario. A dialog box comes up allowing the user to select the target type and to select either an existing path for the new target or to create a new path. If the user chooses to create a new path the user will be taken directly into the edit path dialog box.
(4) Delete	This button removes the active target from the scenario. This action does not delete the target bitmap or other files associated with the target; it just deletes the entry for the target in the scenario description file.
(5) Modify	This button brings up a dialog allowing the user to change the target type for the active target.

The Edit Scenario dialog also has a readout showing the total number of frames for the scenario and button for modifying this number. Hitting the button brings up the dialog which shows the total frame count broken down into two components: "Frames with Movement" and "Trailing static frames." The first of these indicates the length of the movement portion of the longest path currently in the scenario. The remaining frames may be viewed as the pool of frames on which the user can draw in order to add further movement to the longest path. Adjusting the total frame count by keying in a new number effectively adjusts this second number (trailing static frames).

Hitting the Save Changes button at the button of the Edit Scenario dialog causes the EST.DAT file which contains the information for all scenarios to be rewritten.

3.2.3 Editing a Path

The Edit Path dialog is shown in Figure 10. Several user actions cause this dialog box to come up: hitting the edit path button on either the edit scenario dialog box or the edit events dialog box, selecting edit path on the file menu, or choosing to create a new path when adding a new target to a scenario. The upper part of the dialog box lists the primary vertices of the path with fields labeled frame, x, y, view, and frame_ct; these fields hold the starting frame number for the vertex, x and y pixel coordinates, view number (1-48), and frame count. The frame count for a vertex represents the number of frames used in moving from the given vertex to the next primary vertex, except for the last vertex where it represents the number of frames for which the target remains at the last vertex location while the scenario runs to completion. Thus the values in the frame_ct column add up to the total frames for the scenario. As the path is edited, the last vertex frame count is adjusted to maintain that sum, so the last frame count may be regarded as a reserve which the user is drawing down. If the user edits the path in such a way as to cause this reserve to run out, so that the number of frames of movement exceeds the scenario total frames, then the scenario total frames value is increased and a warning message box appears to tell the user that this has happened. In this situation closing the edit path dialog box without saving the results will cause the scenario total frames to be reset to its previous value.

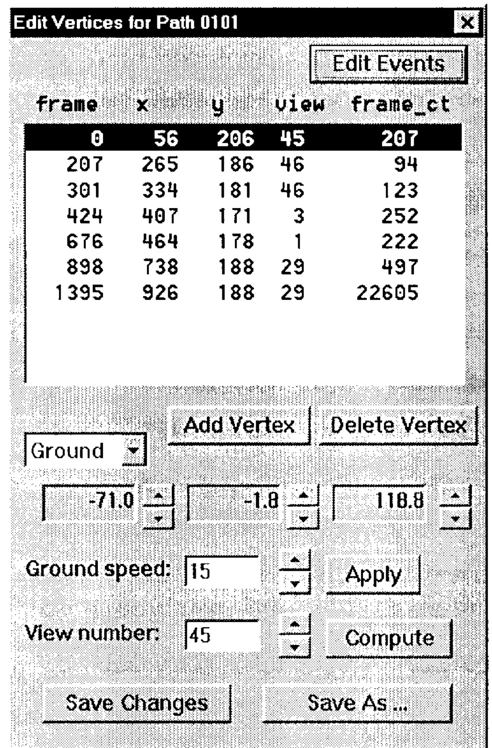

Figure 10: Edit Path Dialog Box

The active target path is displayed in both the main program window and the Top View window, with small boxes indicating the vertices. When the Edit Path dialog is displayed the primary vertices are highlighted (drawn in yellow rather than red) to indicate which ones they are and that they can be edited. The vertex corresponding to the highlighted row in the vertex list is called the active vertex. A vertex may be made active by selecting it in the vertex list, by moving the mouse cursor near one of the primary vertices in the main window, or by selecting it in the Top View window when the Place Vertex command is active. Making a vertex active causes the scenario to move to the starting frame number for that vertex (and pauses the scenario at that point if it was running). Information about the active vertex and the associated path segment is displayed in the fields below the vertex list, and most of the editing options available to the user apply to this vertex or segment. The user may view and edit the vertex location in either ground or pixel coordinates on the Edit Path dialog. The user may edit the ground speed there as

well by entering a value (in meters/sec) and hitting the apply button. When a stationary vertex becomes active, the ground speed field changes to a frame count field, allowing the user to enter the number of frames at which the target is stationary at that location. The view number for a vertex may be changed using the up/down arrows or computed based on the direction of the path segment using the compute button. These changes take place immediately and the display of the target updates to reflect the change.

The active vertex may be moved interactively with the mouse in either the main window or the Top View window, in the standard manner of drag/drop mouse operations using the left mouse button. When this is done with a vertex where one or more stationary vertices are located, the whole set of vertices at that location move together. Frame counts are adjusted for the preceding and/or following segments so as to maintain the ground speeds along those segments at their previous values. This means that moving a vertex typically draws down or adds to the frame count for the last vertex. Other editing operations such as changing a ground speed or in some cases adding or deleting a vertex have this effect as well.

New vertices may be inserted into the path using the Add Vertex button. The new vertex goes after the active vertex and becomes the new active vertex. The user has the choice of adding the new vertex at the same location or at a new location. In the first case, the previously active vertex becomes a stationary vertex. In the second case the new vertex is added at the halfway point along the active segment, or if it is being added at the end of the path, at a location to the right of the last vertex. The active vertex may be deleted using the Delete Vertex button; the path is altered to join the preceding and following vertices.

The Top View window is shown in Figure 11. The two leftmost toolbar buttons of the Top View window correspond to the commands Place Vertex and Center Point. When one or the other of these commands is active the toolbar button is depressed and a mouse button click in the Top View window is regarded as input to the corresponding command. For the Place Vertex command, the user clicks the mouse button near one of the highlighted vertices and holds the mouse button down while dragging the vertex to a new location. Vertices can only be placed within the triangulated region. The (x, z)

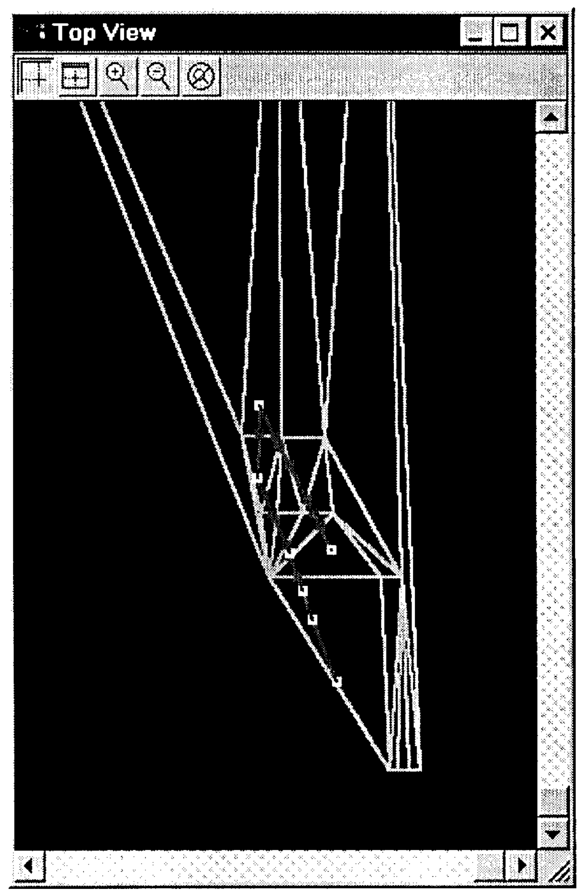

Figure 11: Top View Window

coordinates of the vertex are determined by the cursor location in the Top View window and the y coordinate (elevation) by the triangle in which the vertex falls. If the Center Point command is active when the user clicks inside the Top View window, the location of the mouse click is translated so as to move that location to the center of the window. Since the Zoom In command zooms the window about its center, the Center Point command should be applied first to view a specific location in more detail.

When the path has been edited as desired, the user may save the changes to the current path file (.PTH) using the save button. The path may be saved to a new file based on a new path id using the "save as" button.

References

1. Dow, S. J., *Trainer Methodology Study*, Final Report, Contract DAAH01-91-D-R005 D.O. 75, Sept. 1997.

2. Dow, S. J., *BST Imagery Generation Task*, Final Report, Contract DAAH01-91-D-R002 D.O. 110, Jan. 1998.

3. Preparata, F. P. and Shamos, M. I., *Computational Geometry, An Introduction*, Springer-Verlag, 1985.